(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number : **0 537 903 A2**

(12) **EUROPEAN PATENT APPLICATION**

(21) Application number : **92308416.4**

(22) Date of filing : **16.09.92**

(51) Int. Cl.^{5} : **G06F 15/16**, H04L 12/42

(30) Priority : **02.10.91 US 771064**

(43) Date of publication of application : **21.04.93 Bulletin 93/16**

(84) Designated Contracting States : **DE FR GB**

(71) Applicant : **International Business Machines Corporation**
Old Orchard Road
Armonk, N.Y. 10504 (US)

(72) Inventor : **Douglas, Francis Archibald Brown**
6713 Bonaventure Court
Bethesda, Maryland 20817 (US)
Inventor : **Glowny, David Andrew**
1081 New Haven Road, Apt. 81
Naugatuck, Connecticut 06770 (US)
Inventor : **Mastrangelo, Colette Anne**
24 George Street
Danbury, Connecticut 06810 (US)
Inventor : **Mayer, Paul Melvin**
309 Park Road Extension
Middlebury, Connecticut 06762 (US)
Inventor : **Shier, Peter David**
24 Rodline Road
Danbury, Connecticut 06811 (US)
Inventor : **Shih, Jenngang**
5775 Mosholu Avenue, No.3G
Bronx, New York 10471 (US)
Inventor : **Carr, Brian Patrick**
10D Coach n Four Manor
Jefferson Valley, New York 10535 (US)
Inventor : **Smith, Robin Lynn**
870 Collier Drive
San Leandro, California (US)

(74) Representative : **Killgren, Neil Arthur**
IBM United Kingdom Limited Intellectual Property Department Hursley Park
Winchester Hampshire SO21 2JN (GB)

(54) **Distributed control system.**

(57) A network of distributed workstations is provided for controlling a resource such as a large mainframe computer. Server workstations 108, 110, 112 applying control commands to the resource are attached to each resource 102, 104, 106, preferably through redundant connections 114, 118, 116, 120. Remote client consoles 130 are defined which may be connected to the server for control of that resource. Server and resource location and primary and fallback connection paths are maintained by a centralized control server 160, 162. Upon client console request, the central control server causes the server workstation associated with a particular resource to establish a control session between the server and the client console.

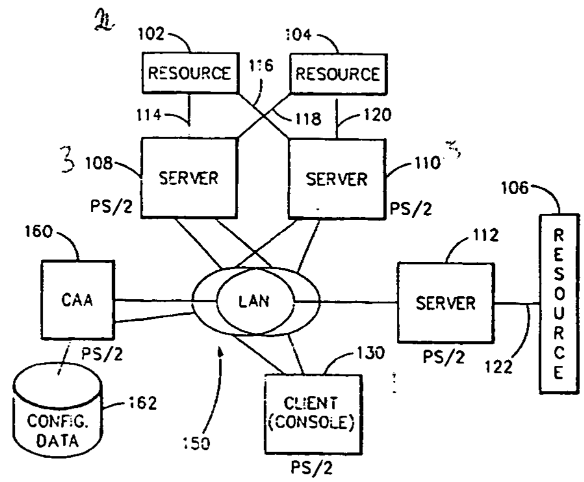

FIG.1

EP 0 537 903 A2

The present invention relates to the management of distributed hardware resources, for example the management of distributed systems that provide remote operation facilities for computer hardware resources.

The increasing power and complexity of large computer systems, frequently termed "mainframe computer systems", has resulted in an increase in the complexity of computer system operations. The drive for increased workforce productivity, however, has tended to reduce the number of personnel assigned to the operations task. The proliferation of large computer system complexes, such as those used by airline reservations systems, banking centers, and similar computer intensive businesses, has also contributed to the need for more effective facilities for the control of hardware systems resources.

Large computer systems have traditionally been operated from an attached console accessible to the computer operators in a computer room. Each computer system has a dedicated console. Thus, in a large complex of, for example, six computers, six operator consoles require monitoring. Each of these computer consoles displays messages in the order generated by the computer system. Many of the messages are informational, indicating the status of certain operations on the computer systems. Other messages provide warnings of current or impending problems. Finally, a third class of message requires operator response to a request for action, such as mounting a tape, or to correct an error detected in the system. It becomes increasingly difficult for an operator to monitor several consoles with several different types of messages and be able to effectively respond to each one.

One solution to the increasing flow of messages is to develop an automated operations facility that is able to interpret and classify messages. These types of facilities can be constructed to segregate messages by message type and to present the operator with those requiring attention. Automated operations facilities of this type are typically constructed using a workstation computer that can be connected directly to the large computer system. The workstation computer contains the necessary programs for receiving, analyzing, and responding to certain messages.

Productivity gains are frequently achieved by centralizing operator resources in a single operations area. This area may be on a different Floor or in a different building than the large computers themselves. Centralization requires that remote access and control of the hardware resource be provided. However, remote access creates a series of problems.

The first problem is the need to allow access to the hardware resource independent of the location of that resource. Next, recovery from the failure of any component in the control system must be possible. In other words, control system component failure must not cause the failure of control of the larger system. Finally, the control system must be flexible allowing the addition of controllable resources and individual control points without disrupting the ongoing control activities.

The problem of remote operations and management has been addressed in several ways. In U.S. Patent Application No. 07/577,967, filed September 4, 1990, commonly assigned, (EPA 478,942) an automated operations system is described which involves a controller coupled to the processor with remote workstation access for controlling that processor. This configuration provides control but limits remote access and fails to address the problem of control system redundancy and reconfiguration.

U.S. Patent No. 5,005,122 suggests the use of a client server model for network management tasks. This system provides for management of a local area network (LAN) through the designation of a single network management node which directs other nodes to perform backup, software distribution, or other network management tasks. While this system provides a means for managing a network, there is no recognition or teaching of the management of a large mainframe hardware resource. In particular, there is no recognition of the requirement to establish fault tolerant connection facilities between a console client and the hardware resource.

Thus, there remains a technical problem of creating a system for remotely controlling a resource such as a large computer system in a manner that allows remote access, failure recovery, and configuration flexibility.

Accordingly, the invention provides a system for distributed control of one or more hardware resources comprising:

network means for establishing communications between a plurality of network elements including one or more server means for issuing control commands to said one or more hardware resources, each of said server means being attached to one or more of said hardware resources, and one or more client means for interacting with an operator and for accepting operator commands to control said one or more hardware resources;

administration means for specifying connection paths to each of said hardware resources from at least one of said client means through said server means for communications therebetween; and

failure recovery means for re-establishing communications between a client means and a server means following failure of a previously established connection path.

Thus the system can establish the location of the resource to be controlled and create a link between a control console and that resource. In addition, the system can recognize and recover from the failure of any control system component. In a preferred em-

bodiment the network means comprises a dual token ring network, to provide a fault tolerant capacity in case one of the token rings is incapacitated.

The invention also provides a method for distributed control of one or more hardware resources in a network including one or more server means for issuing control commands to said one or more hardware resources, each of said server means being attached to one or more of said hardware resources, and one or more client means for interacting with an operator and for accepting operator commands to control said one or more hardware resources, said method comprising the steps of:

specifying network connection paths to each of said hardware resources from at least one client means through a server means, based on central session configuration data stored in a control server;

establishing communications between the client means and the server means in accordance with the specified connection paths; and

recovering from failure by re-establishing communications between a client means and a server means following failure of a previously established connection path.

Preferably the step of establishing communications between a client means and a server means comprises the steps of:

transmitting a session request from said client means to the control server specifying a server means to be connected;

accessing session configuration data in said control server based upon the client and server connection requested;

transmitting configuration data from said control server to said server means; and

establishing a server/client session, said server means initiating said session, based on said fallback configuration data.

It is further preferred that a failure is recovered from by: detecting client control failure in said server means; transmitting a fallback request from said server means to said control server; accessing fallback configuration information in said control server; transmitting fallback information from said control server to said server means; and establishing a server/client session, said server means initiating said establishment based on said fallback configuration data.

Typically the establishment of a connection would result in transmitting a successful session acknowledgement from said client means to said control server; and storing said session acknowledgement in said control server.

It is also preferred that a first task operating on a server means receives messages from the client means, and a second task operating on the server means forwards the messages to a hardware resource, a message being passed from the first task to the second task using an encrypted security key generated by:

generating a check sum value for the data in the message;

copying the check sum value into an addressable memory segment of the server means;

adding a random number to the address of the memory segment to create a modified address;

adding a random number to the check sum value to create a modified check sum;

converting the check sum, modified address and modified check sum into a single string of ASCII digits;

encrypting the string.

Preferably the step of encrypting the string comprises the steps of exchanging bytes within the single string of digits; and encrypting said exchanged bytes using an exclusive or function. Once the key has been generated it is used in a preferred embodiment by:

passing a message between the first and second tasks, said message including said encrypted security key and at least one data field;

validating the security key in the second task by decrypting the key and separating out component elements for verification against expected values;

processing said message if said validating is successful, and

terminating with an error message if validation is not successful.

It should he noted that the security procedure for passing messages between first and second tasks is not limited to distributed control systems, but is of potentially much wider application, and could be used for securely transmitting messages between tasks operating in a computer environment having a processor and a memory.

Thus a computer can be implemented as a system for providing remote control of a resource. The system provides a network for maintaining remote communications between workstations and resources to be controlled. It provides redundant control servers for actually controlling the resource. Client means are connected to the communication network and provide an operator interface and operations control capabilities. Client means further provide an ability to accept operator commands and direct them to the control server as required. Administration means are provided for specifying the locations of resources and potential connection paths and fallback paths for those resources. Finally, failure recovery means are provided for detecting and correcting control system failure by re-establishing client-to-resource connections.

A computer implemented system is therefore used for distributed control of hardware resources. The present system provides a remotely distributed network that allows communication between a series of network components that include servers for con-

trolling a hardware resource and clients for interacting with an operator. Means are also provided for administrating a control network such that connections are established between the elements as required and that any failure of connections is detected and corrected through the use of redundant network components. A record of each dynamic connection is maintained by the central control server. Upon failure of a client console, the server accesses central server fallback information and establishes a fallback connection with an operational client console. Failure of a server workstation causes the resource to establish a link with a backup server which is then able to recover all sessions with client consoles by accessing the dynamic configuration database of the central controller. Configuration data in the control system may be updated dynamically taking effect when the next request for establishment of a session is made.

An embodiment of the invention will now be described by way of example with reference to the following drawings:

Fig. 1 is a block diagram illustrating a system according to the present invention;

Fig. 2 is a block diagram illustrating the steps of establishing an initial client console-to-server session;

Fig. 3 is a flowchart illustrating the process steps for establishing an initial client/sever session;

Fig. 4 is a block diagram illustrating the step of requesting creation of a session between a client and a host resource;

Fig. 5 is a block diagram illustrating the actions occurring upon the failure of a client console workstation;

Fig. 6 is a flowchart illustrating the process steps of recovering from a client failure;

Fig. 7 is a block diagram illustrating recovery from a server failure;

Fig. 8 is a flowchart illustrating the process steps for recovering from a server failure;

Fig. 9 is a block diagram illustrating the administrative update of configuration data; and

Fig. 10 is a flowchart depicting interprocess security key generation.

As shown in Figure 1, a centralised control process for a distributed resource controller manages hardware resources, eg to control the operation of a mainframe computer system such as the IBM 3090 computer system (IBM and 3090 are trademarks of the IBM Corporation). The system can also be used to control other resources which require operator interaction including resources such as a production assembly line, chemical processing facility, or building environmental system. The structure shown in Fig. 1 has resources to be controlled, identified generally at 102, 104, and 106. It will be recognized that this system is applicable to any number of resources and the example of three is shown for illustrative purposes only.

Each resource is connected to at least one server workstation. Server workstations are shown at 108, 110, and 112. These workstations are an intelligent device such as an IBM Personal System/2 computer system (PS/2) (Personal System/2 and PS/2 are trademarks of the IBM Corporation). The servers are connected to hardware resources by communication lines 114, 116, 118, 120 and 122. In the preferred embodiment, a resource typically is connected to more than one server to provide a fallback control path. For example, resource 102 is connected via communication path 114 to server 108 and by communication path 116 to server 110.

A client workstation 130 (such as an IBM PS/2 computer system) is provided as a console for operator use. The client workstation manages the operator interaction including the presentation of messages to the operator and the generation or acceptance of commands for controlling the hardware resource. The separation of client and server functions allows a balancing of workload between the devices. It also allows a single client console to manage a number of hardware resources. In the example shown in Fig. 1, client console 130 can manage resources 102, 104, and 106.

The client console communicates with the servers by means of a local area network (LAN) shown generally at 150. This network can be any one of several known networks, such as the IBM Token Ring LAN, an Ethernet LAN, or others. In a preferred embodiment, the control scheme is implemented using dual token ring LANS which provide for redundancy in case of the failure by either individual LAN.

A central control facility containing control and administrative applications (CAA) is provided to control the operation of the control system. This control server 160 is a workstation which has been configured to support and control the overall operation of the resource control system. The control server provides centralized control to ensure that console access exists for each resource to be controlled and that the client workstations know how to access each resource and how to recover from component failure. Control server 160 manages the network based upon configuration data stored in a database 162. This data can be stored in a variety of known forms such as files on the server or data under the control of the database management system.

The configuration data consists of two major types of data. First, static configuration data contains a description of each hardware resource location to be controlled. It also contains the parameters necessary for accessing and controlling that resource and identifies primary and fallback access paths to that resource. The second type of data is dynamic configuration data describing the current configuration for controlling each resource. This data is maintained to

assure that each resource is under the control of a console and for use by fallback processing routines to re-establish communications in the case of a failure.

The operation of the system to establish a resource control environment will be described with reference to Fig. 2. A resource control environment includes a console for operator interaction, control routines and communications processes for hardware resource management. Fig. 3 is a flowchart presenting the process steps required to establish the control environment.

The control environment for a particular hardware resource is initiated by an operator requesting that a console session be established between a particular client console and a resource. The client console is typically distributed remotely from the hardware resource and a network session must be established between the client console and the resource. The operator requests an access to a resource by selecting that resource from a list of resources presented in a user interface. Fig. 4 illustrates the network accesses required to establish a console session with a particular resource. Screen image 402 is an example of a visual interface to the operator presenting a list of resources that may be selected for console connection. The operator uses a cursor control device, such as a mouse, to select a resource from the menu provided. The requestor program 404 operating in workstation 130 uses the services of a workstation information manager 408 to determine the location of the control server CAA 160. The location of the control database is specified in a data segment 410 and is used by the workstation for access over token ring LAN 150 to the control server 160.

Next (step 302 in Fig. 3) the control server 160 accesses configuration database 162 to determine the location of the hardware resource to be accessed by the console. In the example in Fig. 2, resource 102' is to be accessed. The configuration data 162 will determine that resource 102' is controlled by server 108' and will initiate a session startup 304 by sending a message to server 108', which is responsible for establishing a control session (step 306) with the client console 130. Once the session is successfully established over the network, the client console 130 sends (step 308) a successful session startup acknowledgement to the control server 160. The control server will then record 310 the session startup information in the dynamic data portion of configuration database 162.

Fig. 5 illustrates the message flow for recovering from a client console failure. Fig. 6 is a flowchart describing that process. In step 610, the server 108' detects the failure of the network session between it and the client console 130. Server 108' sends a fallback request to control server 160 (step 612). The control server accesses the configuration database 162 to determine configuration fallback data. Fallback information is sent 616 to server 108' that then establishes a new session 618 with the fallback client console, in this case, console 132. Client console 132 sends an acknowledgement 620 to the control server 160 when the session is successfully established. Control server 160 records the new session information 622 in the dynamic portion of configuration database 162.

Fig. 7 illustrates the process for recovering from the failure of a resource server and Fig. 8 lists the process steps for that recovery. The failure of a server must be detectable by the resource under control 102' which then must have the ability to switch to a backup server. Upon detection of server failure 810, resource 102' initiates control switch to backup server 110'. Server 110' recognizes the resource action and issues 414 a fallback request to the control server 160. Control server 160 accesses the configuration database to determine the client console sessions impacted by the failure of server 108'. The dynamic configuration data indicates all connected sessions and can be used by the fallback server 110' to re-establish those connections. The fallback information is sent 118 to server 110'. Server 110' establishes new connections with each client console previously connected to server 108', for example, client console 130. Upon successful session establishment, client console 130 sends an acknowledgement of the new session 822 to the control server 160. The control server records the new session information in the dynamic portion of the configuration database 162 as step 124.

Interprocess security can be used to enhance the reliability of server tasks. The control server 160 and hardware resource servers (e.g. 108') operate using "frontend" tasks for network communication and "backend" tasks for configuration control and resource control. The division of processing tasks into frontend and backend tasks raises the issue of security in the communications between those frontend and backend tasks. Because the backend tasks interact with computer system resources and thereby have a major impact upon computer system operation, security of messages and data transmitted to the backend tasks is important. The backend tasks must be able to ensure that the messages it receives originated from an authorized frontend task and not another unauthorized program attempting to manipulate the system resources. Second, the backend task must be able to detect possible modifications of data during the transmission process from an authorized frontend to a backend. Finally, the backend must be able to detect the situation where authorized frontend information is captured, modified, and later transmitted to the backend. The solution to this problem is to develop a security key to be combined with the data sent between the front and backends. This key must be such that interception and modification of an au-

thorized message is detected and that it be difficult to decompose the key to reverse engineer the security algorithm.

A security key meeting these requirements is constructed on the basis of three values. First, the address of a shared memory segment sharable between frontend and backend tasks. Second, a standard check sum of the data. Finally, a random number. The combination of these three components provides a secure key meeting the objectives of the security system. First, the address of a shared memory segment indicates the task sending the data is authorized to operate on the computer system and have access to the shared memory. The check sum of the data ensures that the data received by the backend task has not been modified enroute or captured and modified. Finally, the random number introduces a degree of variability and randomness into the key.

The generated key is created according to the process shown in Fig. 10. This operates as follows.

The checksum is first copied into a shared memory segment of known address. The random number is added to the adress of the shared memory segment. The random number is added to the check sum value. All three numbers are converted into ASCII digits. The results of the conversion are concatenated into a single string of digits. The bytes in the concatenated string are exchanged according to a predetermined pattern. The resulting character string is encrypted using a logical "exclusive or" operation on a character-by-character basis with a known static value.

The security key for the message is sent with the data to the backend task. The backend task validates the key by reversing the above construction process.

This method of constructing the key is unique because it combines elements derived from the calling environment (the address of the shared memory segment), along with elements derived from the actual data sent (the check sum value), and a random factor to help conceal the security method and protect against attempts to reverse engineer the key structure algorithm.

Fig. 9 illustrates the update network configuration which allows for dynamically updating the configuration information in configuration database 162. An administrative program operates in a client server, for example, client server 134. (The administrative program could also operate on one of the console client servers, e.g., 130 or 132). The administrative program collects the information necessary to update the configuration data to add a resource, or -to change resource access paths or fallback paths and transmits it for update over token ring LAN 150. In the preferred embodiment, the database update is accomplished using the remote data services facility of the Operating System/2 (OS/2) operating system (OS/2 is a trademark of the IBM Corporation). The update is accomplished by modifying configuration database 162. The modified configuration information will take effect whenever control server 160 is required to establish a new session between client and server workstations. Existing sessions will continue to process as originally set up until manually terminated or until a failure is detected, in which case the above-described failure recovery process will be employed using the updated configuration information.

Claims

1. A system for distributed control of one or more hardware resources (102, 104, 106) comprising:

   network means (150) for establishing communications between a plurality of network elements including one or more server means (108, 110, 112) for issuing control commands to said one or more hardware resources, each of said server means being attached to one or more of said hardware resources, and one or more client means (130) for interacting with an operator and for accepting operator commands to control said one or more hardware resources;

   administration means (160, 162) for specifying connection paths to each of said hardware resources from at least one of said client means through said server means for communications therebetween; and

   failure recovery means for re-establishing communications between a client means and a server means following failure of a previously established connection path.

2. The system of Claim 1 wherein said network means comprises a dual token ring network.

3. A method for distributed control of one or more hardware resources (102, 104, 106) in a network including one or more server means (108, 110, 112) for issuing control commands to said one or more hardware resources, each of said server means being attached to one or more of said hardware resources, and one or more client means (130) for interacting with an operator and for accepting operator commands to control said one or more hardware resources, said method comprising the steps of:

   specifying network connection paths to each of said hardware resources from at least one client means through a server means, based on central session configuration data stored in a control server;

   establishing communications between the client means and the server means in accordance with the specified connection paths; and

   recovering from failure by re-establishing

communications between a client means and a server means following failure of a previously established connection path.

4. The method of claim 3, wherein the step of establishing communications between a client means and a server means comprises the steps of:
transmitting a session request from said client means to the control server (160) specifying a server means to be connected;
accessing session configuration data (162) in said control server based upon the client and server connection requested;
transmitting configuration data from said control server to said server means; and
establishing a server/client session, said server means initiating said session, based on said fallback configuration data

5. The method of claim 4, further including recovering from a failure by:
detecting client control failure in said server means;
transmitting a fallback request from said server means to said control server;
accessing fallback configuration information in said control server;
transmitting fallback information from said control server to said server means; and
establishing a server/client session, said server means initiating said establishment based on said fallback configuration data.

6. The method of claim 4 or 5 further comprising the steps of:
transmitting a successful session acknowledgement from said client means to said control server; and
storing said session acknoledgement in said control server.

7. The method of any of claims 3 to 6, wherein a first task operating on a server means receives messages from the client means, and a second task operating on the server means forwards the messages to a hardware resource, a message being passed from the first task to the second task using an encrypted security key generated by:
generating a check sum value for the data in the message;
copying the check sum value into an addressable memory segment of the server means;
adding a random number to the address of the memory segment to create a modified address;
adding a random number to the check sum value to create a modified check sum;
converting the check sum, modified address and modified check sum into a single string of ASCII digits;
encrypting the string.

8. The method of Claim 7 wherein the step of encrypting the string comprises the steps of:
exchanging bytes within the single string of digits; and
encrypting said exchanged bytes using an exclusive or function.

9. The method of claim 7 or 8, further comprising the steps of:
passing a message between the first and second tasks, said message including said encrypted security key and at least one data field;
validating the security key in the second task by decrypting the key and separating out component elements for verification against expected values;
processing said message if said validating is successful, and
terminating with an error message if validation is not successful.

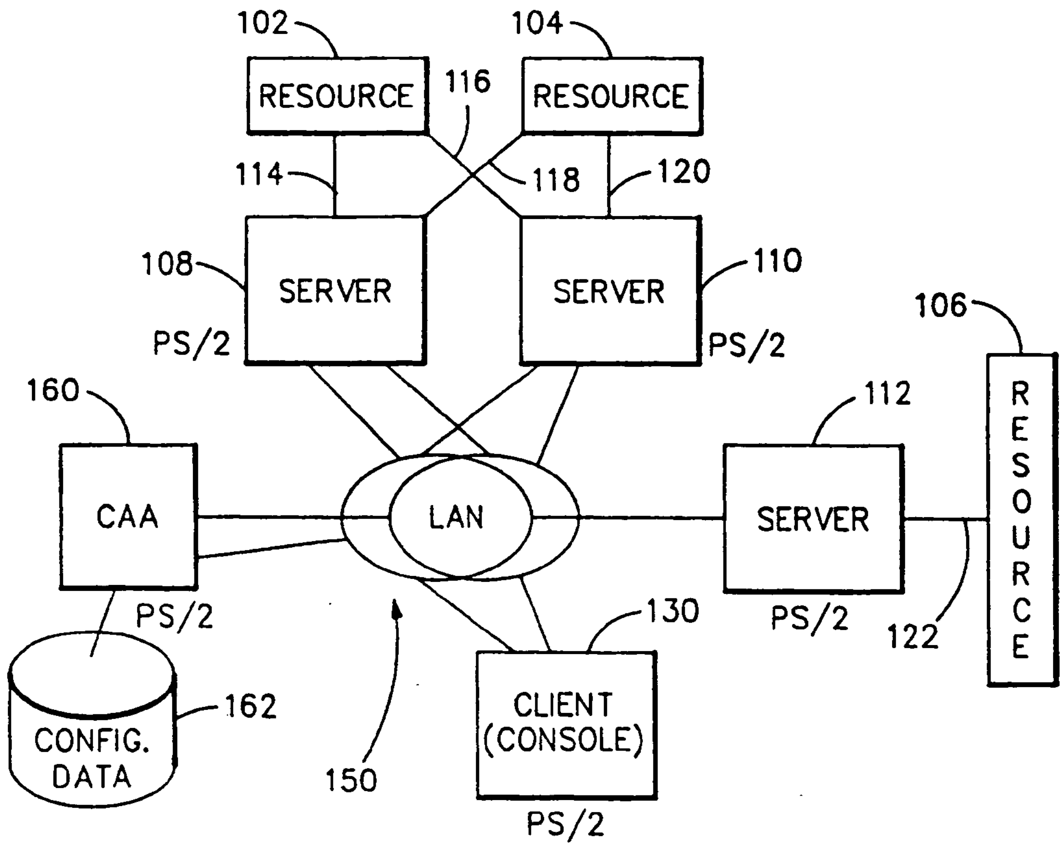
102
104
RESOURCE
116
RESOURCE
114
118
120
108
SERVER
SERVER
110
PS/2
PS/2
106
160
112
R
E
S
O
U
R
C
E
CAA
LAN
SERVER
PS/2
130
PS/2
122
162
CONFIG.
DATA
150
CLIENT
(CONSOLE)
PS/2

FIG.1

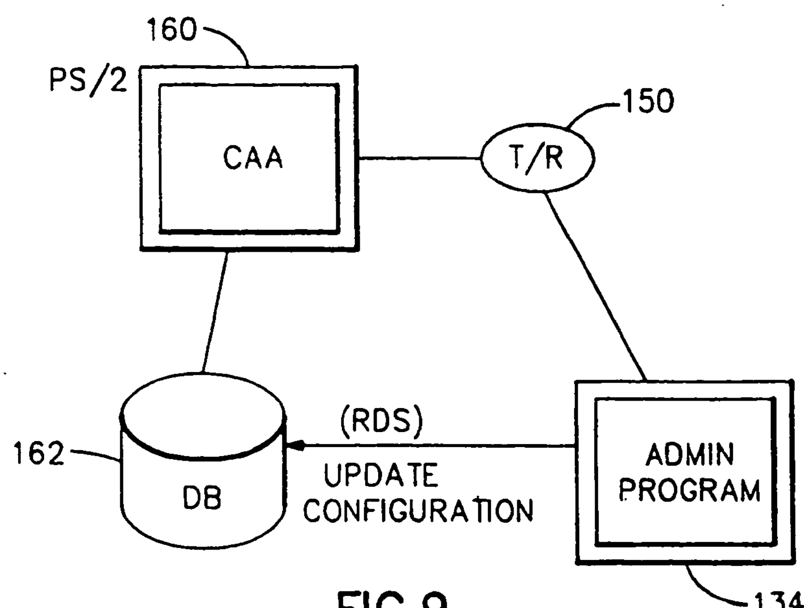
160
PS/2
CAA
150
T/R
(RDS)
162
UPDATE
CONFIGURATION
DB
ADMIN
PROGRAM
134

FIG.9

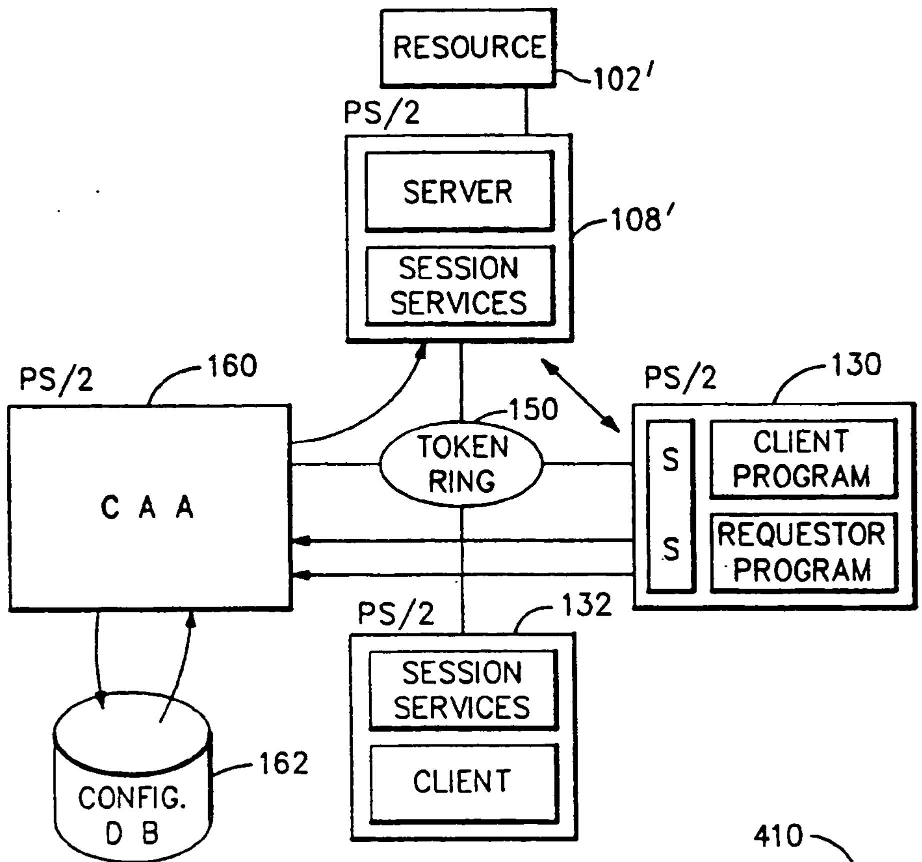
RESOURCE
102'
PS/2
SERVER
SESSION SERVICES
108'
PS/2
160
C A A
150
TOKEN RING
PS/2
130
S
S
CLIENT PROGRAM
REQUESTOR PROGRAM
PS/2
132
SESSION SERVICES
CLIENT
CONFIG. D B
162

FIG.2

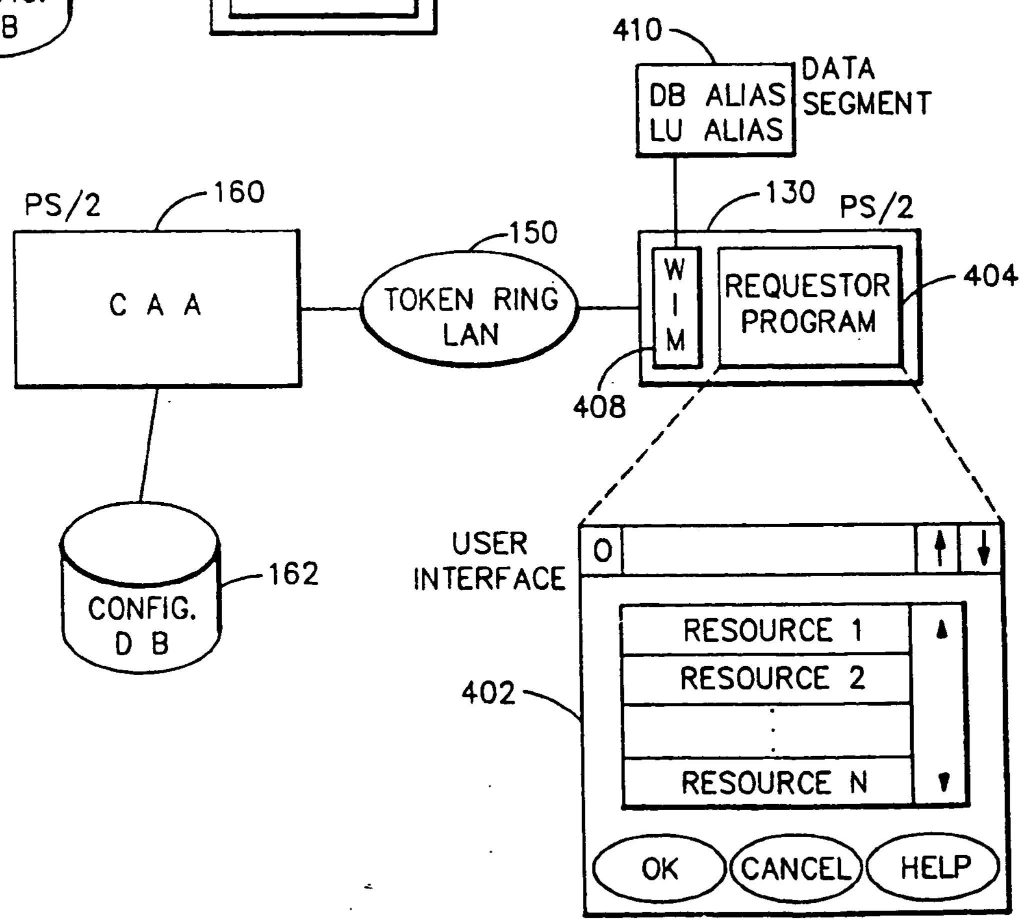
410
DB ALIAS
LU ALIAS
DATA SEGMENT
PS/2
160
C A A
150
TOKEN RING LAN
130
PS/2
W I M
REQUESTOR PROGRAM
404
408
CONFIG. D B
162
USER INTERFACE
402
O
RESOURCE 1
RESOURCE 2
RESOURCE N
OK
CANCEL
HELP

FIG.4

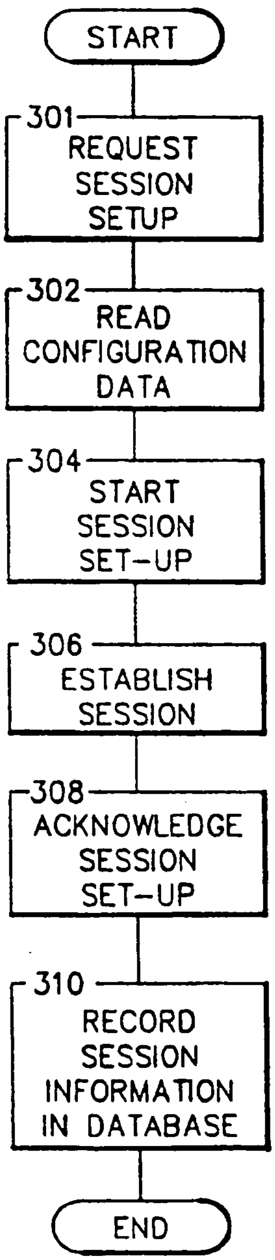

FIG.3

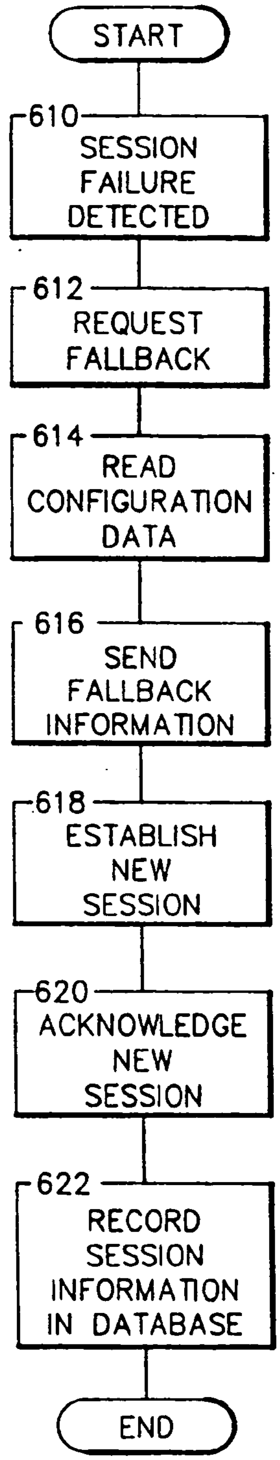

FIG.6

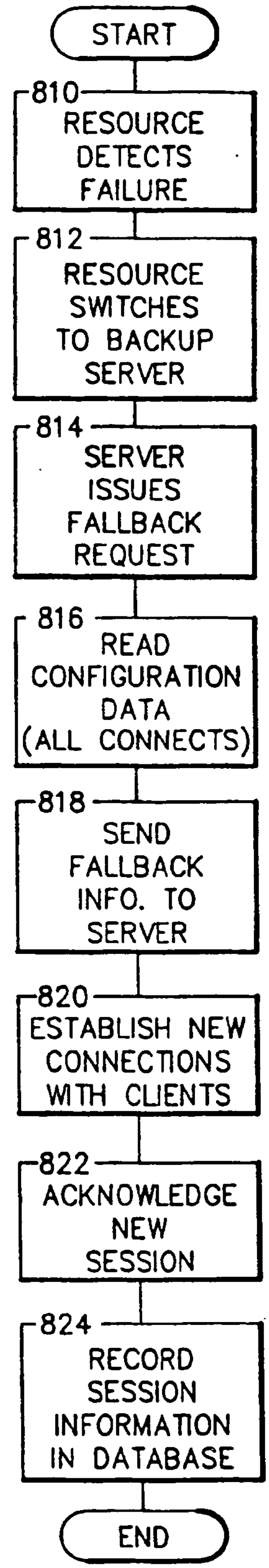

FIG.8

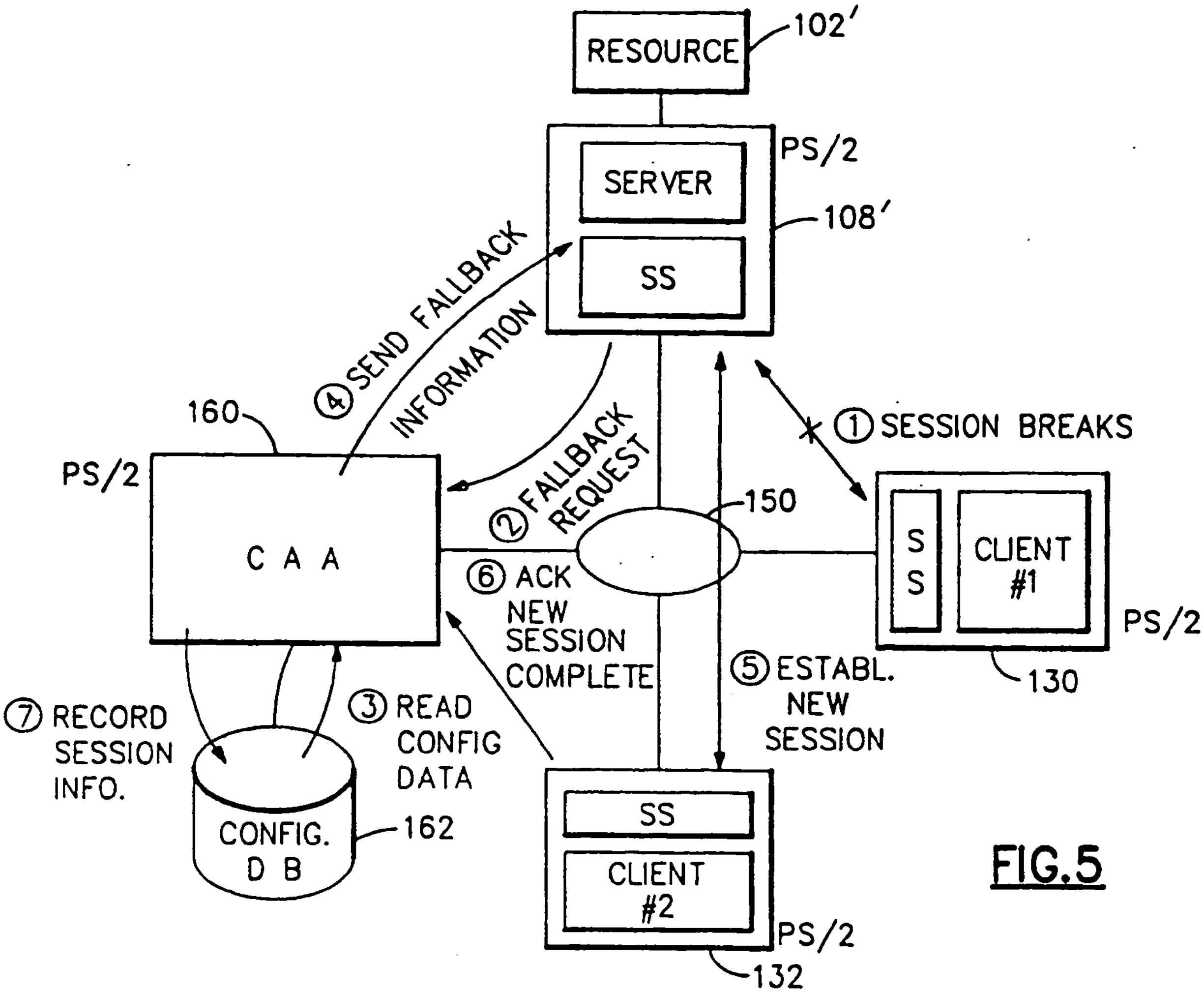
RESOURCE
102'
PS/2
SERVER
108'
SS
④ SEND FALLBACK INFORMATION
160
PS/2
C A A
② FALLBACK REQUEST
① SESSION BREAKS
150
S S
CLIENT #1
PS/2
130
⑥ ACK NEW SESSION COMPLETE
⑤ ESTABL. NEW SESSION
⑦ RECORD SESSION INFO.
③ READ CONFIG DATA
CONFIG. D B
162
SS
CLIENT #2
PS/2
132

FIG.5

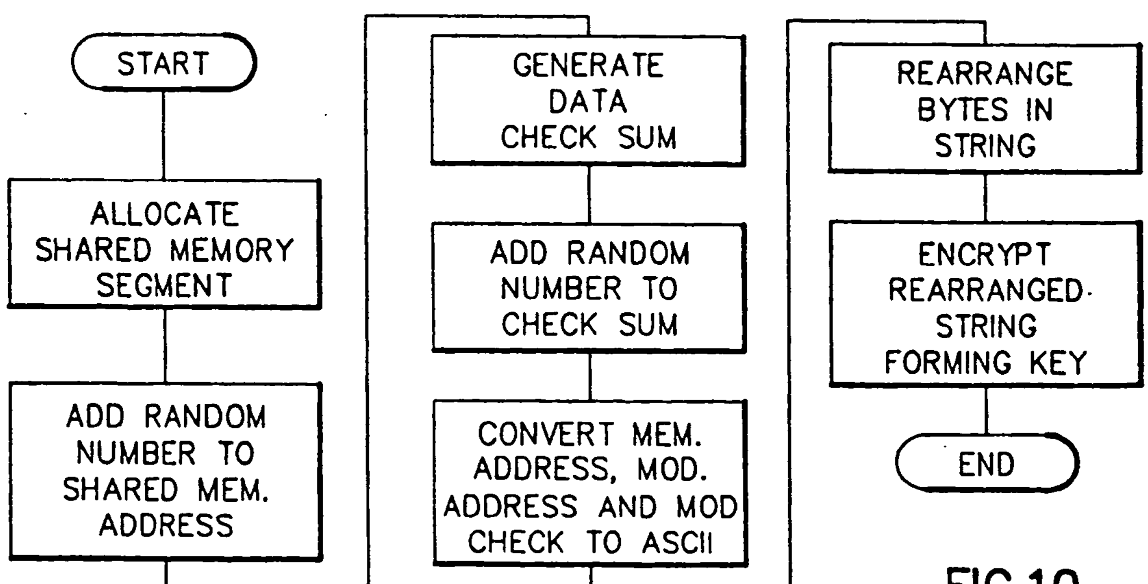
START
ALLOCATE SHARED MEMORY SEGMENT
ADD RANDOM NUMBER TO SHARED MEM. ADDRESS
GENERATE DATA CHECK SUM
ADD RANDOM NUMBER TO CHECK SUM
CONVERT MEM. ADDRESS, MOD. ADDRESS AND MOD CHECK TO ASCII
REARRANGE BYTES IN STRING
ENCRYPT REARRANGED STRING FORMING KEY
END

FIG.10

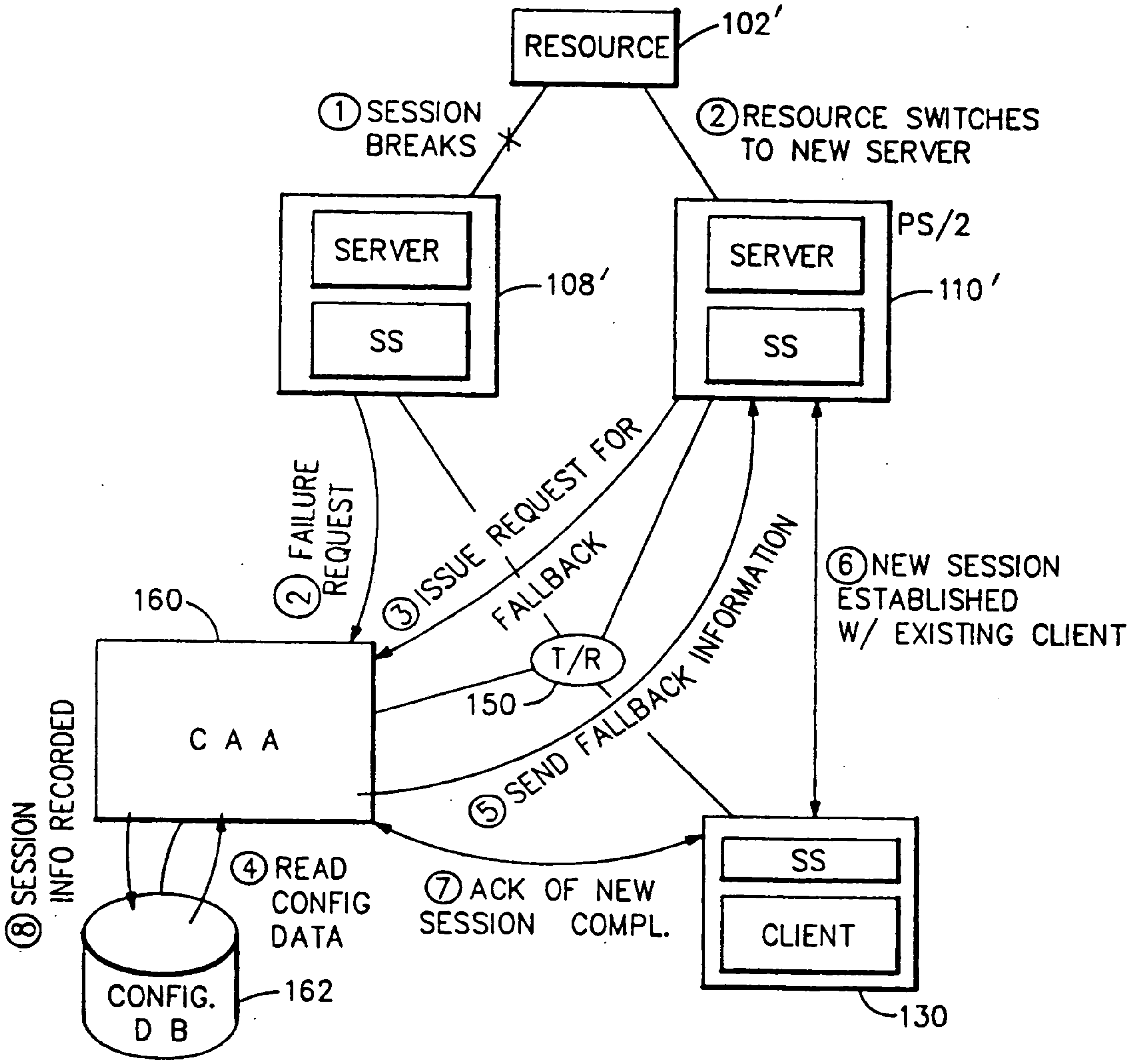
RESOURCE
102′
① SESSION BREAKS
② RESOURCE SWITCHES TO NEW SERVER
SERVER
SS
108′
SERVER
SS
PS/2
110′
② FAILURE REQUEST
③ ISSUE REQUEST FOR FALLBACK
⑤ SEND FALLBACK INFORMATION
⑥ NEW SESSION ESTABLISHED W/ EXISTING CLIENT
160
T/R
150
C A A
⑧ SESSION INFO RECORDED
④ READ CONFIG DATA
⑦ ACK OF NEW SESSION COMPL.
SS
CLIENT
130
CONFIG. D B
162

FIG.7